Star Tracker Attitude Estimation for an Indoor Ground-Based Spacecraft Simulator

Jack Tappe,[1] Jae Jun Kim, [2] Albert Jordan,[3] and Brij Agrawal[4]
Naval Postgraduate School, Monterey, CA, 93943

This paper presents a study of star tracker attitude estimation algorithms and implementation on an indoor ground-based Three Axis Spacecraft Simulator (TASS). Angle, Planar Triangle, and Spherical Triangle algorithms are studied for star pattern recognition. Least squares, QUEST and TRIAD algorithms are studied for attitude determination. A star field image is suspended above TASS. The indoor laboratory environment restricts the placement of the star field to be in close proximity to TASS. This restriction adds some additional complication to the standard attitude determination problem. An iterative solution handles this complication. Experimental verification is also performed for the proposed iterative solution.

I.   Introduction

THE objective is to develop a star tracker precision attitude estimation system for use on an indoor, ground-based spacecraft simulator. Star pattern recognition algorithms are studied with a focus on accuracy and algorithmic efficiency. Attitude determination algorithms are studied similarly. The Three Axis Spacecraft Simulator (TASS) is the testbed and it is equipped with a CCD camera to capture the star field image suspended above.

The star pattern recognition algorithm takes the camera image of the star field, assigns to it a mathematical description of the pattern, and finds the unit vector to each imaged star. Stars in the CCD image appear distributed among multiple pixels and the centroid of this distribution must be found. Unit vectors from the focus point of the camera lens to the image plane of the CCD are then found for each star. These vectors are mapped from the camera reference frame to the spacecraft body reference frame.

The pattern for the stars in the camera image is then found. The pattern is checked against a database of the entire sky to find a match. The pattern can be defined as simply as an angle. Triangles provide more information for more robust matching.[3,4] Novel methods such as grids can also be robust and efficient.[11] Algorithms used in this study are the triangle, planar triangle, and spherical triangle.

Once the star vectors measured in the spacecraft frame are matched to the inertially referenced database of star patterns and star vectors all the necessary information is available to solve the attitude determination problem. Attitude determination algorithms determine the rotation of the star vectors from the inertial referenced frame to the spacecraft body frame. There are many algorithms for attitude determination, but three were studied in this report: least-squares[2], Quaternion-Estimator (QUEST)[8], and the TRIAD[8] algorithm.

II.   Star Pattern Recognition

A.   Angle Method

The angle method is the simplest star identification algorithm. Star pairs are observed by the camera, and their unit vectors are developed referenced to the frame of the star tracker. The star tracker then calculates the angles between all stars within the FOV of the camera. The angles are calculated by the equation 1:

$$\theta = \cos^{-1}\left(r_1 \cdot r_2\right) \qquad 1$$

where r_1 and r_2 are the unit vectors pointing to each star. The angle θ will be the same from the inertial frame as it is viewed from the satellite.

The angle of the stars in the camera's FOV are calculated using Equation 1. However, the angle calculated is in the frame of the star tracker camera. The angles must be converted to the body frame of the satellite for use in any attitude determination algorithm. Those body frame angles must be compared to angles in the inertial reference

		Form Approved
Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **AUG 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**
4. TITLE AND SUBTITLE **Star Tracker Attitude Estimation for an Indoor Ground-Based Spacecraft Simulator**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Monterey,CA,93943**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**
13. SUPPLEMENTARY NOTES
14. ABSTRACT
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**19**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

frame. Therefore, an onboard database of inertial stars with their angles calculated from the inertial reference frame must be available.

B. Planar Triangle

Another algorithm for star identification is the method of planar triangles[3]. The star tracker develops a triangle from a combination of three stars. The benefit of this algorithm is that more information can be obtained from a triangle than an angle, which should allow the star tracker to determine the satellite's attitude more accurately.

From the calculated triangle, the area and polar moment can be determined. The area and polar moment provide two additional pieces of information to match in the database. By observing three stars with unit vectors $\tilde{b}_1$, $\tilde{b}_2$, and $\tilde{b}_3$, the area of the planar triangle is obtained by:

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

$\qquad$ 2

where

$$s = \frac{1}{2}(a+b+c)$$

$\qquad$ 3

$$a = \left\| \tilde{b}_1 - \tilde{b}_2 \right\|$$

$\qquad$ 4

$$b = \left\| \tilde{b}_2 - \tilde{b}_3 \right\|$$

$\qquad$ 5

$$c = \left\| \tilde{b}_1 - \tilde{b}_3 \right\|$$

$\qquad$ 6

The above equations are valid in the Earth-Centered-Inertial or ECI frame as well as the star tracker frame. In the planar triangle method, three observed stars provide far more information than only two stars using the angle method. As shown, there are multiple angle calculations as well as other features of the triangle to store.

It will also be necessary to obtain the polar moment in conjunction with the area of the triangle. Two triangles may have the same area, but will have different second moments. The polar moment by:

$$J = \frac{A\left(a^2 + b^2 + c^2\right)}{36}$$

$\qquad$ 7

When using planar triangles, the use of the triangles polar moment and planar area will reduce the number of similar solutions for matching, however there are certain costs with using this algorithm. There are more features a triangle can provide when compared to an angle. Naturally, instead of determining the satellite's position with two stars, it now requires three stars if using the planar triangle algorithm. There are however more mathematical calculations that must be performed when using the triangle algorithm compared to the angle calculations. Also, with the triangle there are two data points for each triangle which will require a larger memory to hold this data.

C. Spherical Triangle

A similar algorithm used in star pattern recognition is spherical triangles[4]. The star tracker calculates a spherical triangle when it observes three stars within its FOV. Again, the polar moment and area are used to determine which spherical triangle is being observed by the star tracker.

The three unit vectors to the stars within the FOV, allow the area of the spherical triangle to be calculated by:

$$A = 4\tan^{-1}\sqrt{\tan\left(\frac{s}{2}\right)\tan\left(\frac{s-a}{2}\right)\tan\left(\frac{s-b}{2}\right)\tan\left(\frac{s-c}{2}\right)}$$

$\qquad$ 8

where S is the same as above and a, b, and c are:

$$a = \cos^{-1}\left(\frac{b_1 \cdot b_2}{|b_1||b_2|}\right)$$

$\qquad$ 9

$$b = \cos^{-1}\left(\frac{b_2 \cdot b_3}{|b_2||b_3|}\right)$$

$\qquad$ 10

$$c = \cos^{-1}\left(\frac{b_3 \cdot b_1}{|b_3||b_1|}\right) \qquad\qquad 11$$

Again, the above equations are valid in the ECI frame as well as the star tracker frame. The polar moment is also valuable information to be obtained from each observed triangle. Two similar triangles may have similar areas or polar moments, but it is unlikely that two triangles will have exactly the same polar moments and areas. The acquisition of two unique pieces of information from each triangle makes the algorithm resistant to false attitude determinations from the star tracker using an erroneous triangle.

The polar moment of a triangle is obtained by breaking the spherical triangle into smaller triangles. The area of each of these smaller triangles is then multiplied by the square of the arc distance from the centroid of each smaller triangle, to the centroid of the overall triangle. The spherical triangle's polar moment is then obtained by summing the results of each smaller triangle:

$$I_p = \sum \theta^2 dA \qquad\qquad 12$$

where dA is the smaller triangle area and θ^2 is the arc distance. The polar moment of each spherical triangle is calculated via a recursive algorithm that breaks the triangle into smaller triangles successively until the depth of recursion is met.

III.   Attitude Determination

The problem of attitude determination is obtaining the correct orthogonal rotation matrix, so that the measured observations in the sensor frame match the reference frame. The measured vectors are the aforementioned body-frame vectors to imaged stars while the reference vectors are those same stars referenced from the ECI frame and contained in the database.

The stars imaged in the FOV of the star tracker have now been paired to stars in the inertial frame by the star pattern recognition algorithms, but the attitude of the spacecraft is still unknown.

For this section, the inertial reference unit vectors are represented by $\hat{V}_1 \dots \hat{V}_n$, and the body frame unit vectors are represent by $\hat{W}_1 \dots \hat{W}_n$. Therefore, an orthogonal matrix A is needed that satisfies:

$$A\hat{V}_i = \hat{W}_i, \qquad (i = 1, \dots, n) \qquad\qquad 13$$

Due to measurement errors and corruption in both the star tracker measurements and errors in the inertial vectors, there is no exact solution for A. Therefore an approach is needed to select an A that matches $\hat{V}_i$ to $\hat{W}_i$. This is known as Wahba's Problem[12].

Wahba's problem is the estimation of a satellite's attitude by using direction cosines. Given two sets of points, in this case $\hat{V}_1 \dots \hat{V}_n$ and $\hat{W}_1 \dots \hat{W}_n$ where $n \geq 2$, find a rotation matrix A which aligns the first set of vectors into the best least squares coincidence with the second set of vectors. Mathematically, a matrix A minimizes:

$$\sum_{j=1}^{n} \left\| \hat{W}_j - A\hat{V}_j \right\|^2 \qquad\qquad 14$$

where $\|\cdot\|$ denotes the Euclidean norm. This equation can be represented in the terms of a cost or loss function as:

$$L(A) = \frac{1}{2}\sum_{i=1}^{n} a_i \left| \hat{W}_i - A\hat{V}_i \right|^2 \qquad\qquad 15$$

subject to the constraint:

$$AA^T = I_{3x3} \qquad\qquad 16$$

The quadratic loss function in the attitude matrix can be transformed into a quadratic loss function in the corresponding quaternion[9]. Wahba, presents a least-squares criterion to define the best estimate for an orthogonal matrix A that minimizes the cost function represented by Equation 15.

There are many different types of attitude determination algorithms for star trackers in use today, but a common type used is a class that estimates the four Euler symmetric parameters that form the quaternion[13]. The quaternion outputs of these algorithms are extremely popular as it is the minimal non-singular set for global attitude description.

The quaternion also provides an attitude matrix which is quadratic in the parameters and is also free of transcendental trigonometric functions[2]. The optimal estimator of the quaternion can be used to solve the constrained least-squares Wahba problem[13].

Other algorithms used in solving Wahba's problem by obtaining the quaternion are the TRIAD[6] algorithm as well as the Quaternion Estimator (QUEST)[8] algorithm. The TRIAD and QUEST algorithms each provide quaternions as well as the direction cosine matrix of the satellite. The TRIAD algorithm is fairly simplistic while without requiring any inversion of matrices while the QUEST algorithm requires fairly complex eigenvalue calculations.

A. Least Squares

The star tracker camera will have some noise which will cause errors in the measurements. To account for errors, most of the error is concentrated on a small area about the direction of Ar_i, and therefore the sphere containing that point is approximated as a tangent plane, which is represented by the following equation[4]:

$$\tilde{b}_i = Ar_i + v_i,$$
$$v_i^T Ar_i = 0$$

<div align="right">17</div>

Here $\tilde{b}_i$ is the ith measurement and the sensor error $\mathbf{v}_i$ is approximately Gaussian[4]. Therefore all angle measurements will contain some error and this error must be accounted for.

The error or residual errors are assigned to each measurement of r_i. Therefore, Equation 17 becomes:

$$\tilde{b}_i = A\tilde{r}_i + \tilde{e}_i$$
$$\tilde{b}_i = \begin{bmatrix} \tilde{b}_1 & \tilde{b}_2 & \ldots & \tilde{b}_n \end{bmatrix}$$
$$\tilde{e}_i = \begin{bmatrix} \tilde{e}_1 & \tilde{e}_2 & \ldots & \tilde{e}_m \end{bmatrix}$$
$$\tilde{r}_i = \begin{bmatrix} \tilde{r}_1 & \tilde{r}_2 & \ldots & \tilde{r}_m \end{bmatrix}$$

<div align="right">18</div>

where b_i are the measured values for the inertial star vectors and e_i are the residual errors for each star tracker measurement of r_i.

Using Gauss's principle of least squares, it is desired to obtain an A that minimizes the residual errors. Solving for the residual errors we obtain:

$$\tilde{e}_i = A\tilde{r}_i - \tilde{b}_i$$

<div align="right">19</div>

A cost function of residual errors is[2]:

$$J = \frac{1}{2} e^T e$$

<div align="right">20</div>

Or:

$$J = \frac{1}{2} \left(\tilde{b}^T \tilde{b} - 2\tilde{b}^T A\tilde{r} + \tilde{r}^T A^T A\tilde{r} \right)$$

<div align="right">21</div>

There are two requirements for minimizing this quadratic function: 1) a necessary condition and 2) a sufficient condition. The necessary condition and sufficient conditions are defined as[2]:

$$\nabla_r J = \begin{bmatrix} \dfrac{\partial J}{\partial \tilde{r}_1} \\ \vdots \\ \dfrac{\partial J}{\partial \tilde{r}_n} \end{bmatrix} = A^T A\tilde{r} - A^T \tilde{b} = 0$$

<div align="right">22</div>

$$\nabla_r^2 J \equiv \frac{\partial^2 J}{\partial r \partial r^T} = A^T A$$

<div align="right">23</div>

where A^TA must be positive definite. Above, $\nabla_r J$ is the Jacobian and $\nabla_r^2 J$ is the Hessian. The matrix A is positive definite when the matrix has a maximum rank (n). The quadratic function J is a performance surface in n + 1 dimensional space with a convex shape of an n-dimensional parabola with a single distinct minimum[2].

From the necessary conditions defined above, the "normal equations" are:

$$\left(A^T A\right)\tilde{r} = A^T \tilde{b} \qquad\qquad 24$$

If there are n independent observation equations, therefore the rank of A is n, making $A^T A$ positive definite[2]. With equation $A^T A$ positive definite, $\left(A^T A\right)$ is invertible and an explicit solution for the optimal solution is obtained. Therefore $\tilde{r}$ is solved by:

$$\tilde{r} = \left(A^T A\right)^{-1} A^T \tilde{b} \qquad\qquad 25$$

Equation 25 is the matrix equivalent of Gauss' original "equations of condition" in index/summation notation[2]. Naturally, an inverse of $A^T A$ must exist to find a solution for $\tilde{r}$. The inverse exists only if there number of linearly independent observations is equal to or greater than the number of unknowns. In least squares, the order of the matrix inverse is equal to the number of unknowns, not the number of measurement observations[2]. An example of attitude determination with a star will illustrate these principles.

Assume the camera of the star tracker observes two stars within its FOV. The unit vectors of these stars in the star tracker reference frame are designated $\tilde{r}_1$ and $\tilde{r}_2$. These two stars have unit vectors in the inertial frame as well, $\hat{R}_1$ and $\hat{R}_2$. For this example, we will only use one of the stars for calculations. The inertial coordinates of the star are matched the body coordinates by a direction cosine matrix A. Therefore the equation is

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \qquad\qquad 26$$

Rearranging, the equation becomes:

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{pmatrix} a_{11}r_x & a_{12}r_y & a_{13}r_z \\ a_{21}r_x & a_{22}r_y & a_{23}r_z \\ a_{31}r_x & a_{32}r_y & a_{33}r_z \end{pmatrix} \qquad\qquad 27$$

Rearranging further to take the following form:

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix}\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \qquad\qquad 28$$

Equation 28 now takes the form of the normal matrix equation:

$$\hat{y} = A\hat{x} \qquad\qquad 29$$

with

$$
\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \hat{y}, \quad
\begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix} = A, \hat{x} =
\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix}
\qquad 30
$$

The $\hat{y}$ vector is the known inertial coordinates to the star, the A matrix is the known body frame vector to the same star, with the $\hat{x}$ comprising the elements of the direction cosine matrix being the only unknown quantity. Now Equation 29 can be solved by inserting the elements of the matrix equation into the following to form the least squares problem:

$$ \hat{x} = \left(A^T A \right)^{-1} A^T \hat{y} \qquad 31 $$

The vector $\hat{x}$ of the direction cosine matrix is simply reshaped into the usual form to get the direction cosine matrix.

B. TRIAD

The TRIAD algorithm is a deterministic solution that generates a direction cosine matrix between two coordinate systems when two vectors are given in each of the particular coordinate systems[8]. Applying this algorithm to the attitude determination problem is fairly straightforward. The star tracker needs only to see two stars within its FOV to determine two unit vectors. These are referred to as the observed vectors[8]. The other two unit vectors, or reference vectors, are found using the angle, planar triangles, or spherical triangles algorithms defined previously.

Using the TRIAD algorithm, two non-parallel unit vectors to stars in the inertial frame as well as two non-parallel unit vectors in the star tracker frame are obtained. Using the same designation as above, these vectors are identified as $\hat{V}_1$ and $\hat{V}_2$ for inertial stars with two body frame vectors from the star tracker as $\hat{W}_1$ and $\hat{W}_2$. The algorithm then finds an orthogonal matrix A which becomes the attitude matrix for the satellite finds the orientation difference between the two systems[8]. The equations that the algorithm must satisfy are:

$$ A\hat{V}_1 = \hat{W}_1 \qquad A\hat{V}_2 = \hat{W}_2 \qquad 32 $$

The algorithm then requires computation of the following column matrices or triads[8]:

$$ \hat{r}_1 = \hat{V}_1 \qquad \hat{r}_2 = \frac{\left(\hat{V}_1 \times \hat{V}_2 \right)}{\left| \hat{V}_1 \times \hat{V}_2 \right|} $$

$$ \hat{r}_3 = \frac{\left(\hat{V}_1 \times \left(\hat{V}_1 \times \hat{V}_2 \right) \right)}{\left| \hat{V}_1 \times \hat{V}_2 \right|} $$

33

$$\hat{s}_1 = \hat{V}_1 \qquad \hat{s}_2 = \frac{\left(\hat{W}_1 \times \hat{W}_2\right)}{\left|\hat{W}_1 \times \hat{W}_2\right|}$$

34

$$\hat{s}_3 = \frac{\left(\hat{W}_1 \times \left(\hat{W}_1 \times \hat{W}_2\right)\right)}{\left|\hat{W}_1 \times \hat{W}_2\right|}$$

There exists a unique orthogonal matrix that satisfies:

$$A\hat{r}_i = \hat{s}_i \quad (i = 1, 2, 3)$$

35

which is defined as:

$$A = \sum_{i=1}^{3} \hat{s}_i \hat{r}_i^T$$

36

The triads are then constructed into matrices for further computation. A reference matrix is made consisting of the reference triads while an observed matrix is likewise constructed of observed triads. The matrices are:

$$M_{ref} = \left[\hat{r}_1 \vdots \hat{r}_2 \vdots \hat{r}_3\right] \qquad M_{obs} = \left[\hat{s}_1 \vdots \hat{s}_2 \vdots \hat{s}_3\right]$$

37

where M_{ref} and M_{obs} matrices are 3x3 matrices. The attitude determination matrix is obtained by:

$$A = M_{obs} M_{ref}^T \quad \text{or}$$
$$A = r_1 \cdot s_1^T + r_2 \cdot s_2^T + r_3 \cdot s_3^T$$

38

There are problems with the TRIAD algorithm though. The first vector has more prominence in determination of A. Some of the information in the second vector is discarded[8]. It is therefore necessary and best practice to obtain use the most accurate instrument to find the first vector of each set, in this case $\hat{V}_1$ and $\hat{W}_1$. Therefore, the first vector, or anchor vector, may be obtained by the star tracker while the second vector cold come from the magnetometer[8].

C. QUEST

The QUEST algorithm[8] developed for the Magsat mission by Shuster is another method to solve Equation 32. The quadratic loss in the attitude matrix function can be converted to a corresponding quaternion. The result is that an eigenvalue equation is obtained that provides the quaternion [8]. This result is that the optimal quaternion is computed by a fast deterministic algorithm.

Equation 32, the loss function, is minimized when an optimal matrix A_{opt} is determined, however, we can also maximize a gain, g, that also solves the same equation. In Equation 32, the nonnegative a_i, $i = 1, \ldots, n$ are a set of weights [8]. Since the loss function may be scaled without affecting the resultant A_{opt}, it is thereofre possible to set:

$$\sum_{i=1}^{n} a_i = 1$$

39

The corresponding gain function $g(A)$ is given as

$$g(A) = 1 - L(A) = \sum_{i=1}^{n} a_i \hat{W}_i^T A \hat{V}_i = q^T K q$$

40

It is easy to see that the loss, $L(A)$, function will be at a minimum when the gain function, $g(A)$ is at its maximum [8]. This can be interpreted in the following way as well:

$$g(A) = \sum_{i=1}^{n} a_i tr\left[\hat{W}_i^T A \hat{V}_i\right]$$

41

where *tr* represents the trace operation performed in MATLAB. The matrix A is usually represented as quaternions since they are simpler to use.

To continue with this algorithm, several other quantities will need to be calculated to form the matrix K. The matrix is a 4x4 matrix that takes the following form:

$$K = \begin{pmatrix} S - \sigma I & Z \\ Z^T & \sigma \end{pmatrix}$$

42

where Z is a 3x1 vector, $S - \sigma I$ is a 3x3 matrix, Z^T is a 1x3 matrix, and σ is a scalar [8]. The matrix S is defined from the equation:

$$S = B + B^T = \sum_{i=1}^{n} a_i (\hat{W}_i \hat{V}_i^T + \hat{V}_i \hat{W}_i^T)$$

where

43

$$B = \sum_{i=1}^{n} a_i \hat{W}_i \hat{V}_i^T$$

The vector, Z, is defined as:

$$Z = \sum_{i=1}^{n} a_i \left(\hat{W}_i \times \hat{V}_i \right)$$

44

The quantity σ is the trB or:

$$\sigma = \sum_{i=1}^{n} a_i \hat{W}_i \cdot \hat{V}_i$$

45

Using these quantities, the gain function can be written in the following form:

$$g(\tilde{q}) = \left(q^2 - Q \cdot Q \right) trB^T + 2tr \left[QQ^T B^T \right] + 2q tr \left[\underline{Q} B^T \right]$$

46

where

$$\underline{Q} = \begin{pmatrix} 0 & Q_3 & -Q_2 \\ -Q_3 & 0 & Q_1 \\ Q_2 & -Q_1 & 0 \end{pmatrix}$$

47

Using the matrix K, this produces a bilinear equation of the form:

$$g\left(\tilde{q} \right) = \tilde{q}^T K \tilde{q}$$

48

Using the original constraint, the quaternion that maximizes can be used by implementing Lagrange multipliers [8]. A new gain function is defined. Using the notation of introduced by Shuster and Oh, this gain function is denoted as $g'\left(\tilde{q} \right)$. The gain function is written as:

$$g'\left(\tilde{q} \right) = \tilde{q}^T K \tilde{q} - \lambda \tilde{q}^T \tilde{q}$$

49

which is maximized without constraint [8]. The variable, λ, is used to satisfy this constraint. The verification is satisfied by differentiating which produces the equation:

$$K\tilde{q} = \lambda \tilde{q}$$

50

Therefore, the optimal quaternion is an eigenvector of the matrix K, and λ is an eigenvalue. The maximizing of $g'\left(\tilde{q} \right)$ will occur by choosing the eigenvector that corresponds to the largest eigenvalue of the matrix K [8]. Therefore:

$$K\tilde{q}_{opt} = \lambda \tilde{q}_{opt}$$

51

IV. Experimental Setup

A. Three Axis Spacecraft Simulator

The second-generation Three Axis Spacecraft Simulator (TASS) at the Naval Postgraduate School's Spacecraft Research and Design Center was used for the experimental setup. TASS floats on an air-bearing to simulate space flight and provide three rotational axes of motion. TASS is five feet in diameter. It uses four control moment gyroscopes for actuation. Onboard computers can execute Matlab code and Simulink models real time with xPC Target. Attitude sensors include the star tracker, inertial measurement unit, sun sensor, magnetometer, inclinometer, and a precision laser sensor.

Suspended one meter above TASS is an LCD monitor displaying a star field. Since the star tracker camera cannot be located at the center of rotation of TASS the reference frame fixed to the star tracker will not only be rotated but also translated. In space, this translational motion of the frame is not observable since stars are approximately infinitely far away. In a laboratory environment this translational motion will affect the star tracker algorithms.

Figure 1. Three Axis Spacecraft Simulator testbed with star tracker camera sticking out the top. LCD monitor suspended in the ceiling displays the star field image.


Figure 2: Star field suspended above TASS.

Figure 3 illustrates the star field, the inertial frame I centered at the center of rotation of the spacecraft simulator, the camera fixed frame B, the translated inertial frame I' and the translated body frame B'. The center of the translated inertial frame I' moves with the camera, but does not rotate and its axes are parallel to those of frame I. The translation of frame I from frame I' is done by position vector $\tilde{R}_i$. Frame B is the frame associated with the camera and therefore is fixed with the camera. The frame B' is translated from frame B such that the origin of frame B' coincides with the origin of the inertial frame I.

For star fields located at a far distance, the unit vectors represented in frame B are related to the unit star vector $\hat{r}_i$ represented in frame I by the following relationship:

$$b_i^B = Ar_i^I \qquad\qquad 52$$

where A is the direction cosine matrix representing the attitude of the spacecraft simulator and the superscripts denote the specific frame used for the star vectors. In addition, the angle between b_1 and b_2 is same as the angle between r_1 and r_2 for distant stars. When the stars are displayed in close proximity, however, the angle between b_1 and b_2 is not same as the angle between r_1 and r_2 as can be seen in Figure 3.
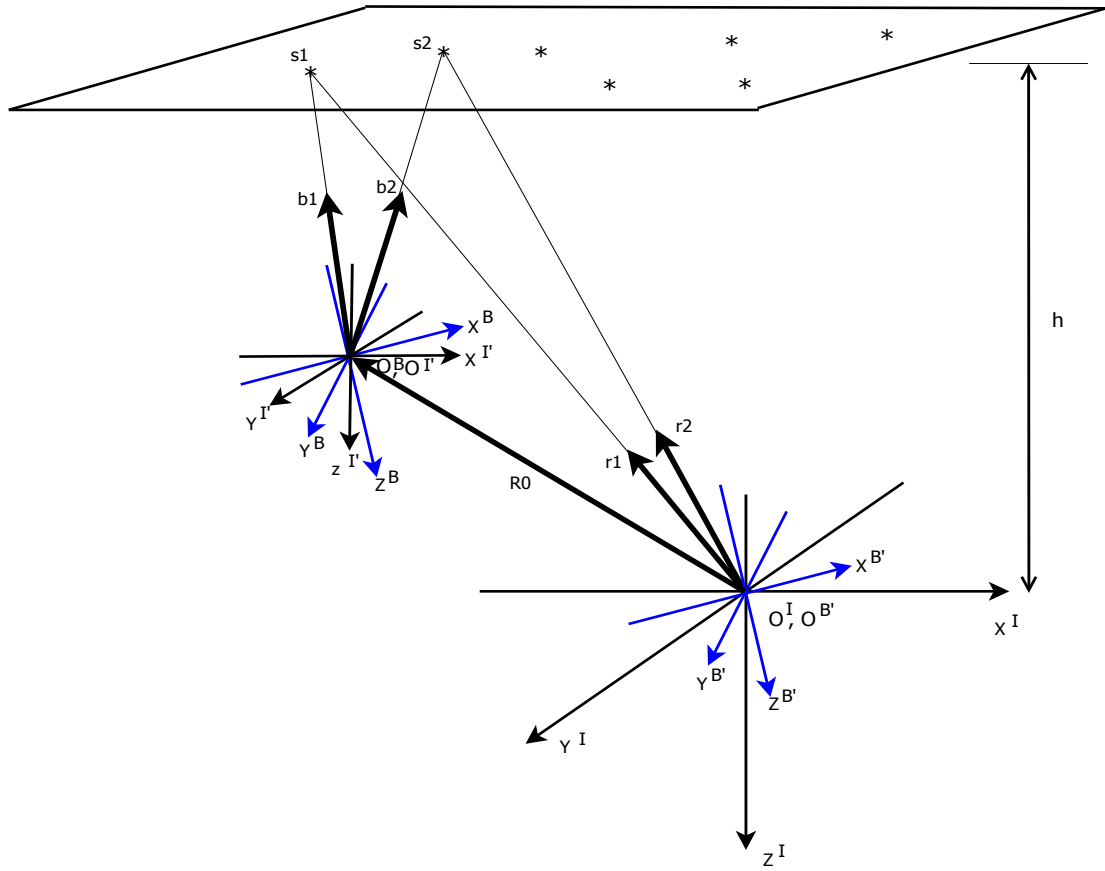
Figure 3: Vector representations of star field, star tracker frame, and the inertial frame.

In order to solve this problem, let us first define α_i representing the distance between ith star and the origin of frame B (or frame I'). Similarly, we can also define β_i being the distance between ith star and the origin of frame I (or frame B'). Using α_i and β_i, the following relationship can be found.

$$R_0 + \alpha_i b_i = \beta_i r_i \qquad\qquad 53$$

which can be rewritten as:

$$R_0^{B'} + \alpha_i b_i^B = A\beta_i r_i^I \qquad\qquad 54$$

From above, $R_0^{B'}$ is a constant vector fixed to the spacecraft body, b_i^B is measured by the star tracker camera, and r_i^I is a star vector represented in the inertial frame which will serve as the database. The equation is not a linear equation to solve for the attitude matrix because α_i is a function of A.

Assuming that the star tracker camera is looking at the $-z^B$ direction and defined a vector $p^{I'} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$. The distance from the origin of the inertial coordinate system to the monitor is 1.7695 m which is defined as h. The α is derived from:

$$\alpha_i \left(\hat{b}_i^{I'} \cdot \hat{p}^{I'} \right) = h - \hat{p}^{I'} \tilde{r}^I \qquad\qquad 55$$

Equation 55 can be rearranged so that:

$$\alpha_i = \frac{h - \left(\hat{p}^{I'} \right)^T A^T \tilde{r}_o}{\left(\hat{p}^{I'} \right)^T A^T \tilde{b}_i^B} \qquad\qquad 56$$

The β_i can also be computed as

$$\beta_i = \frac{h}{\left(p^{I'}\right)^T r_i^I} \qquad\qquad 57$$

For reference database, the star vectors measured in frame *B* at zero attitude need to be converted into star vectors in frame *I* . The unit star vector at the inertial frame *I* which will serve as a database becomes

$$r_i = \frac{R_o^{B'} + \alpha_i b_i^B}{\beta_i} \qquad\qquad 58$$

The Figure 4 illustrates an image from the database stars with the respective star numbers. Once, the vectors are obtained in inertial frame *I*, the angle database is completed. All the inertial angles are calculated. The inertial numbers of the stars used to calculate the angles are stored with their respective angles to create a lookup table. The entire database is stored as a MAT file in the TASS computer system. The final database comprises angles with star numbers and the table with the inertial unit vectors and star number.
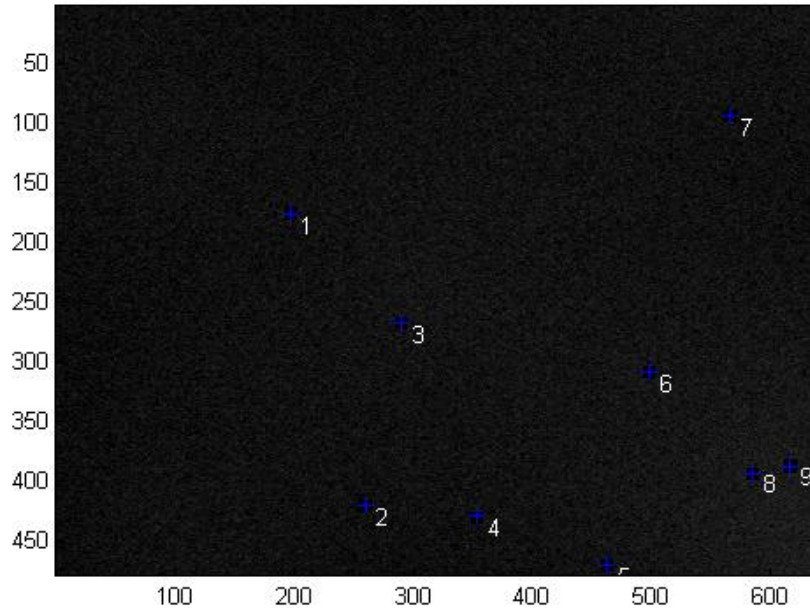

Figure 4: Inertial star database image with numbering.

B. Iterative Attitude Estimation

Equation 54, $R_0^{B'} + \alpha_i b_i^B = A\beta_i r_i^I$, is not easily applicable for computation of attitude matrix *A* because α_i is also a function of *A*. Instead of solving complex optimization problem, we want to apply the algorithms presented in Section III. In order for this, a simple iterative approach is proposed. First, compute the vectors $R_0^{B'} + \alpha_i b_i^B$ using the prediction of a spacecraft attitude *A*. The prediction of a spacecraft attitude can be made from previous estimates of the spacecraft attitude or using additional sensors such as rate gyros. With this attitude prediction, the sect of vectors, $R_0^{B'} + \alpha_i b_i^B$, can be used to compute the angles between them. These angles are now compared with the angles in the database and set of matching stars are identified for attitude estimation. If the prediction of the attitude is not accurate, the accuracy for the angle matches need to be relaxed. In order for accurate

estimation, the resulting attitude estimation can again be used to perform more accurate matches and attitude estimation. Therefore, this method needs several iterations with slight increase in computational power.

V. Results

Experiments were performed to test the fidelity of attitude estimation the angle algorithm. For verification of the results, simple angle method with the QUEST attitude determination algorithm was implemented on the test-bed. Star unit vectors translated to the B' frame from the star image are first computed, and the angles between the brightest or master star and all other stars are calculated. These angles are compared to the inertial angles stored in the database. The experiment is setup so that the prediction of A attitude matrix used for computation of $R_0^{B'} + \alpha_i b_i^B$ has either no errors or some errors while the test-bed is at zero attitude. By doing these tests, we can measure the accuracy of the estimation as well as the required accuracy of the prediction of the A matrix for the proposed algorithm.

After the angles are matched, the star inertial vectors and observed star vectors are then entered into the QUEST algorithm. The QUEST algorithm will then calculate an updated or accurate A matrix and attitude quaternion. This new A matrix can then be used as a new initial estimate for further attitude calculation iterations.

A. Estimation Results without Iteration

For these tests, there is only one iteration of attitude updates. A series of A matrices with an initial error of 6 degrees, 3 degrees, 0 degrees, -3 degrees, and -6 degrees were chosen. A matching accuracy of 500 arc-seconds for each angle was chosen for all A matrices. Therefore, an angle from an observed star angle must match an inertial angle by the value ± 250 arc-seconds. All the multiple matches were filtered out to ensure accurate results in the experiment. For each test with 50 attitude estimations, several parameters were observed. The resulting average updated A matrix from the QUEST algorithm for each test with its standard deviation were recorded. The number of stars and angles matched were saved, as well as the Euler angles and their standard deviations.

a) Testing with an A of 0 degree error.

With a prediction of the A matrix of 0 degree error, in other words a 3x3 identity matrix, five test runs with 50 attitude determinations for each test were accomplished with a 500 arc-second accuracy. The results of the attitude testing for the Euler angles and their standard deviations are included in Table 1. Even though there is no error inserted, only seven stars out of nine are easily matched due to the noise of the system. The mean values and standard deviations for all the Euler angles remain fairly constant over the five tests as well.

Table 1.   Tabulated results for Euler angles and standard deviation for an A matrix of 0 radian error.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	6.86E-04	5.65E-04	6-004	5.65E-04	9.28E-04	6.86E-04

Theta	3.70E-03	3.00E-03	2.00E-03	3.00E-03	5.00E-03	3.34E-03
Psi	2.97E-05	2.30E-04	1.30E-04	2.30E-04	4.30E-04	2.10E-04

σ Phi	1.20E-03	1.10E-03	9.18E-04	1.10E-03	1.40E-03	1.14E-03
σ Theta	0.0066	0.0061	0.005	0.0061	0.0074	6.24E-03
σ Psi	6.74E-04	6.18E-04	5.06E-04	6.18E-04	7.57E-04	6.34E-04

Number of stars matched	7.000	7.000	7.000	7.000	7.000	7.000

The average attitude matrix for this testing is:

$$A = \begin{pmatrix} 0.9943 & 0.0002 & 0.0027 \\ 0.0003 & 0.9927 & 0.0006 \\ 0.0033 & 0.0005 & 0.9983 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0121 & 0.0006 & 0.0065 \\ 0.0006 & 0.0154 & 0.0011 \\ 0.0062 & 0.0012 & 0.0035 \end{pmatrix}$$

As seen from these simulations, the attitude is close to the true attitude (identity matrix) due to no errors in the predicted attitude and most of the stars are picked up by the star tracker. The next phase of testing is with an error in the attitude prediction.

b.   *Testing with an A with 3 degrees error.*

The next phase of testing inserted an error of 3 degrees into the Euler angles to form an initial *A* matrix used in attitude determination. Five more test runs for attitude determinations were accomplished with a 500 arc-second matching accuracy. The results of the attitude testing for the Euler angles and their standard deviations are included in Table 2. The QUEST and angle algorithms do correct for the error, but fewer stars are matched. The main thing noticeable from this test is the marked decline in the number of stars detected by the algorithm. Only 3.26 stars are detected due to the initial error during 50 runs. If the initial attitude estimate is off, the results from the angle algorithm decline sharply.

Table 2.   Tabulated results for Euler angles and standard deviation for an *A* matrix of 3 radian error.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	1.70E-03	6.58E-04	2.90E-03	3.50E-03	2.70E-03	2.29E-03
Theta	7.40E-03	1.70E-03	1.36E-02	1.71E-02	1.26E-02	1.05E-02
Psi	3.30E-04	-2.41E-04	9.58E-04	1.30E-03	8.54E-04	6.40E-04
σ Phi	3.70E-03	1.58E-04	5.50E-03	6.20E-03	5.40E-03	4.19E-03
σ Theta	0.0199	4.18E-04	0.0299	0.0335	0.0294	2.26E-02
σ Psi	2.10E-03	5.77E-05	3.10E-03	3.40E-03	3.00E-03	2.33E-03
Number of stars matched	3.260	3.320	3.200	3.320	3.200	3.260

The average attitude matrix for this testing with a 3 radian error is:

$$A = \begin{pmatrix} 0.9832 & 0.0007 & 0.0073 \\ 0.0012 & 0.9786 & 0.0023 \\ 0.0131 & 0.0020 & 0.8006 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0445 & 0.0139 & 0.1072 \\ 0.0144 & 0.0556 & 0.0357 \\ 0.1205 & 0.0896 & 0.0188 \end{pmatrix}$$

c.   Testing with an A with a 6 radian error.

Further testing with a large error of 6 radians was attempted with a matching accuracy of 500 arc-seconds. This error would test the ability of the algorithms to arrive at the correct attitude solution with a large initial estimate error. The results of this test were poor. To get the algorithm to function, the accuracy had to be dropped to 900 arc-seconds or 0.0044 radians to get two angles to match with the database correctly. Four angles were detected, but only two were accurate. The only way to make the algorithm work with this amount of error, is to filter the database further for angles that are within 0.0044 radians of each other.

d.   Testing with an A with a -3 radian error.

The next series of tests involved using an A matrix with an Euler error of -3 radians. The error was inserted and the simulations ran. Table 3 contains the results of the -3 angle error tests. The matching accuracy is maintained at 500 arc-seconds for all tests.

Table 3.   Tabulated results for Euler angles and standard deviation for an A matrix with -3 radian error.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	-3.71E-04	-2.92E-04	-5.49E-05	-2.91E-04	-7.65E-04	4.30E-03
Theta	7.51E-04	5.86E-04	9.27E-05	5.85E-04	1.60E-03	1.06E-02
Psi	-8.86E-05	-7.20E-05	-2.26E-05	-7.23E-05	-1.71E-04	1.20E-03

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
σ Phi	1.20E-03	1.10E-03	5.59E-04	1.10E-03	1.60E-03	9.10E-03
σ Theta	0.0025	0.0023	0.0012	0.0023	0.0033	2.48E-02
σ Psi	2.50E-04	2.26E-04	1.17E-04	2.26E-04	3.33E-04	3.50E-03

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Number of stars matched	6.020	6.000	6.000	6.000	6.000	6.004

From the results in Table 3, the Euler Angles are accurately calculated, but the star matches drops from the nine matches achieved with a zero radian error. Only six of the stars are matched in most of the cases with an initial error. The average attitude matrix for this testing with a -3 radian error is:

$$A = \begin{pmatrix} 0.9939 & -0.0001 & 0.0014 \\ -0.0001 & 0.9937 & -0.0006 \\ 0.0012 & -0.0006 & 0.9997 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0196 & 0.0004 & 0.0040 \\ 0.0004 & 0.0202 & 0.0020 \\ 0.0041 & 0.0019 & 0.0011 \end{pmatrix}$$

e.   *Testing with an A with a -6 radian error.*

With an *A* matrix of -6 radians error, five test runs for attitude determination were accomplished with a 500 arc-second accuracy. The results of the attitude testing for the Euler angles and their standard deviations are included in Table 8. The error still allows six stars to match in all cases.

Table 4.   Tabulated results for Euler angles and standard deviation for an A matrix with -6 radian error.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	-6.86E-04	-4.49E-04	-6.08E-04	-4.49E-04	-6.07E-04	4.30E-03
Theta	1.40E-03	9.12E-04	1.20E-03	9.13E-04	1.20E-03	1.06E-02
Psi	-1.55E-04	-7.20E-05	-1.38E-04	-1.05E-04	-1.38E-04	1.20E-03

σ Phi	1.50E-03	1.30E-03	1.50E-03	1.30E-03	1.50E-03	9.10E-03
σ Theta	0.0032	0.0027	0.003	0.0027	0.0015	2.48E-02
σ Psi	3.20E-04	2.71E-04	3.06E-04	2.71E-04	3.06E-04	3.50E-03

Number of stars matched	6.000	6.000	6.000	6.000	6.000	6.000

The average attitude matrix for this testing with a -6 radian error is:

$$A = \begin{pmatrix} 0.9939 & -0.0001 & 0.0014 \\ -0.0001 & 0.9937 & -0.0006 \\ 0.0012 & -0.0006 & 0.9997 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0125 & 0.0003 & 0.0026 \\ 0.0003 & 0.0130 & 0.0013 \\ 0.0026 & 0.0012 & 0.0007 \end{pmatrix}$$

B. Estimation Results with Iterations

The next testing incorporated using the QUEST algorithm to provide an iterative attitude matrix update the attitude estimation. The updated attitude solution should increase the accuracy by providing by providing corrected A matrices as initial estimate to the algorithms matching the inertial database to the body frame image.

The update testing was completed with an initial error of 2 radians in the TASS initial attitude estimate with an matching accuracy requirement of 500 arc-seconds. The testing starts at a single update, and then continues on up to five attitude updates. Theoretically, at each update, the attitude solution will improve. Table 5 contains the results of the testing.
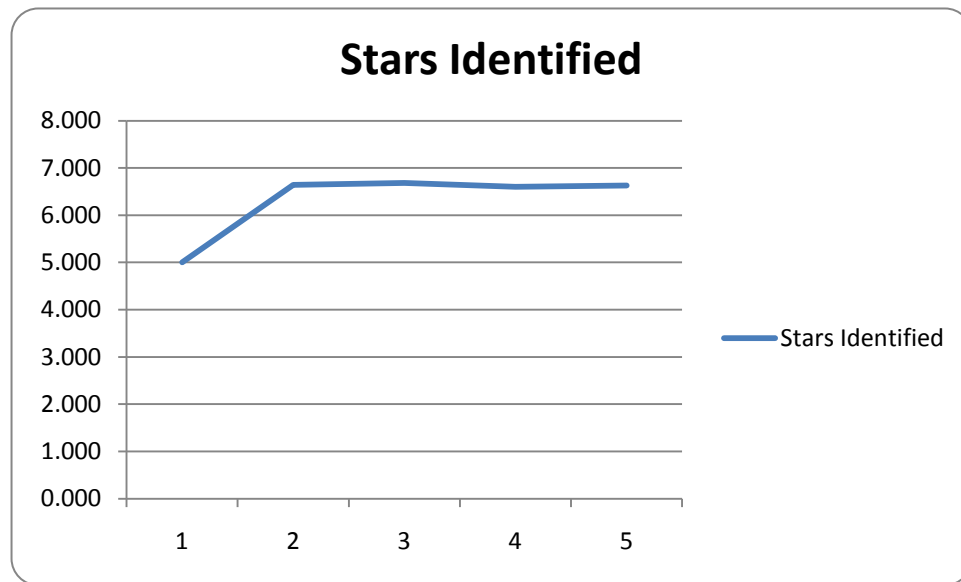
Table 5.   Tabulated results for Euler angles and standard deviation with *A* matrix updates.

	Update 1	Update 2	Update 3	Update 4	Update 5	Overall Average
Phi	1.30E-03	7.98E-04	1.50E-03	2.40E-03	1.30E-03	4.30E-03
Theta	6.50E-03	3.20E-03	4.00E-03	3.10E-03	4.00E-03	1.06E-02
Psi	5.05E-04	3.13E-04	-1.33E-04	-1.63E-04	-6.44E+00	1.20E-03

σ Phi	2.30E-03	1.70E-03	2.90E-03	3.40E-03	2.90E-03	9.10E-03
σ Theta	0.0125	0.0094	0.0087	0.0073	0.0093	2.48E-02
σ Psi	1.30E-03	9.91E-04	1.10E-03	1.30E-03	1.10E-03	3.50E-03

Number of stars matched	5.000	6.640	6.680	6.600	6.630	6.310

The main result of the update test is the increasing accuracy of the initial estimate of the attitude solution. With an initial error of 2 radians, the updates remove the error and provide an updated attitude matrix. As shown in the bottom line, the attitude updates improve with increasing amounts of attitude solution updates. The amount of star matches improves from 5 at the beginning to ~6.6 stars at the end with five updates. Table 5 shows the increase in star identification with increasing updates, while Table 6 shows the Euler angles over the same updates.

By using an updated attitude matrix with errors removed while holding accuracy constant, star matching improves rapidly. With only one update, only five star matches are achieved. When using two updates, the matches increase to almost seven stars identified, which is very close to the testing in Table 1 which was conducted using an *A* matrix of 0 error.

Figure 5: Tabulated results for stars recognized and standard deviation for an A matrix with a 2 radian error.

With increasing A matrix updates, more precise Euler angles should also be obtained from the more accurate A matrix. The Euler angles, shown in Table 11, overall exhibit similar trends of increasing accuracy. Most improved are the Theta and Psi angles
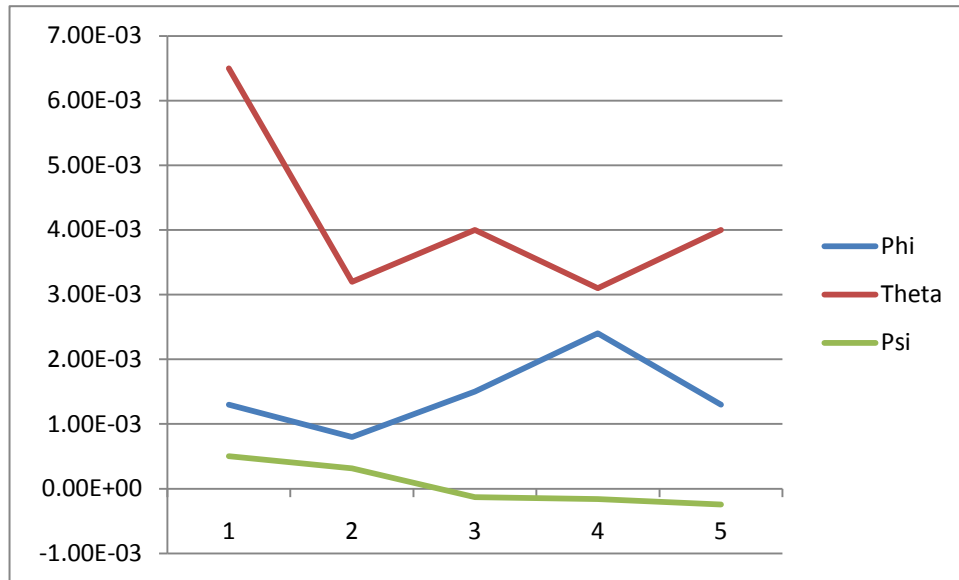
Figure 6: Phi, Theta, and Psi angles over iterations.

VI. Conclusion

The angle algorithm for star pattern recognition and the QUEST algorithm for attitude determination were successfully implemented on the TASS. The algorithm worked with an accuracy of 500 arc-seconds with an initial estimate of position as a 3x3 identity matrix. The algorithms accurately determined the TASS position for a range of error from 3 radians to -6 radians. Beyond 3 radians and -6 radians the angle method breaks down in its ability to accurately determine the position of the TASS.

By updating the A matrix by outputs from the QUEST algorithm, the accuracy of the attitude solution increased markedly until a certain point, then leveled off. The maximum star recognition increased from ~5 stars with one update and to 6.6 stars and leveled off. The testing shows that increased accuracy is obtained by providing an updated attitude solution as an initial estimate to the algorithms.

References

[1]Tappe, J. A., "Development of Star Tracker System for Accurate Estimation of Spacecraft Attitude," M.S. Thesis, Mechanical and Astronautical Engineering Dept., Naval Postgraduate School, Monterey, CA, 2009.

[2]Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems,* Chapman & Hall/CRC, Boca Raton, Florida, 2004, Chaps. 1-7.

[3]Cole, C. L. and Crassidis, J. L., "Fast Star-Pattern Recognition Using Planar Triangles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 64-71.

[4]Cole, C. L., and Crassidis, J. L., "Fast Star Pattern Recognition Using Spherical Triangles," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August 16-19, 2004, Providence, Rhode Island.

[5]Wie, B., *Space Vehicle Dynamics and Control*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA., 1998, Chaps. 5-7.

[6]Lerner, G.M., "Three-Axis Attitude Determination," *Spacecraft Attitude Determination and Control*, edited by J.R. Wertz, Kluwer Academic Publishers, Dordrecht, Holland, 1978, pp. 420-488.

[7]Kuipers, J. B., *Quaternions and Rotation Sequences,* Princeton University Press, Princeton, New Jersey, 1999, Chaps. 1-13.

[8]Wie, B., Weiss, H., and Arapostathis, A., "Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations," Journal of Guidance, Control, and Dynamics, Vol. 12, No. 3, 1989, pp. 375-380.

[9]Shuster, M., and Oh, S., "Three-Axis Attitude Determination from Vector Observations," Journal of Guidance and Control , 70–77.

[10]van Bezooijen, R. W. H., "Automated Star Pattern Recognition," Ph.D. Dissertation, Aeronautics and Astronautics Department, Stanford University, Stanford, CA, 1989.

[11]Padgett, C., Kreutz-Delgado, K., "A Grid Algorithm for Autonomous Star Identification," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 1, Jan. 1997, pp. 202-213.

[12]Wahba, G., Problem 65-1, A Least Squares Estimate of Satellite Attitude. *Society for Industrial and Applied Mathematics*, 1966, pp. 385–386.

[13]Weiss, C. H., Bar-Itzhack, I. Y., & Oshman, Y., "Quaternion Estimation From Vector Observations Using a Matrix Kalmann Filter," *AIAA Guidance, Navigation, and Control Conference and Exhibit.* San Francisco, CA.

American Institute of Aeronautics and Astronautics